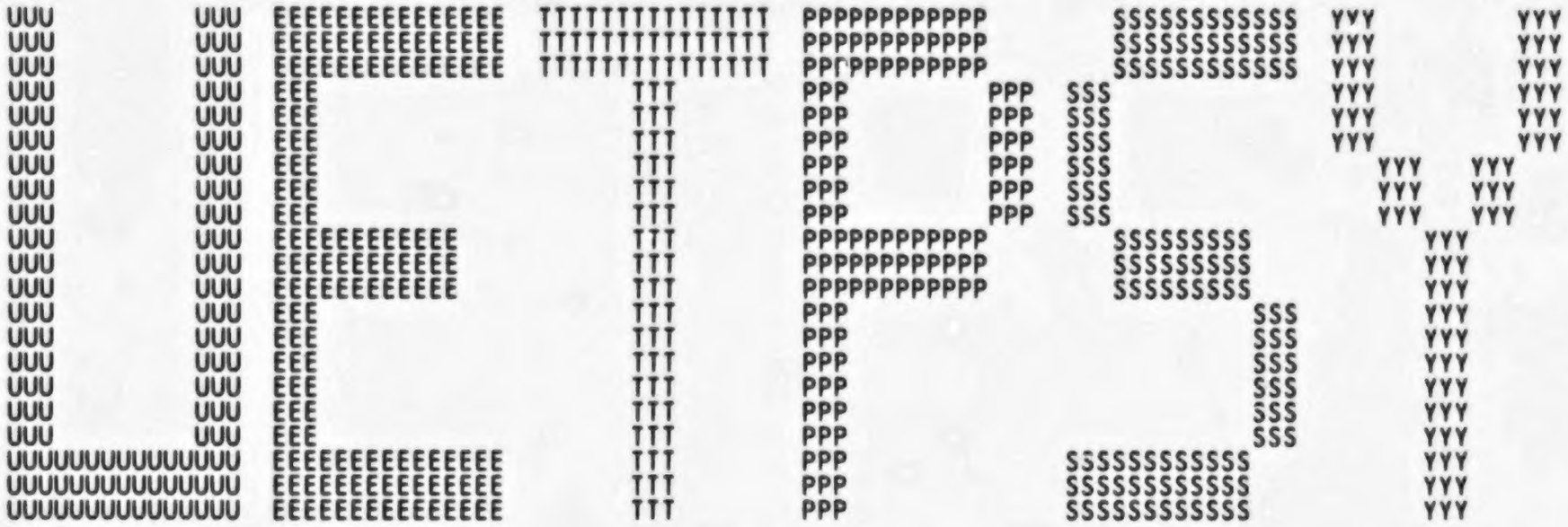
UUU UUU EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT PPPPPPPPPPPP SSSSSSSSSSS YYY YYY
UUU UUU EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT PPPPPPPPPPPP SSSSSSSSSSS YYY YYY
UUU UUU EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT PPPPPPPPPPPP SSSSSSSSSSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEEEEEEEEEE TTT PPPPPPPPPPPP SSSSSSSSS YYY
UUU UUU EEEEEEEEEEE TTT PPPPPPPPPPPP SSSSSSSSS YYY
UUU UUU EEEEEEEEEEE TTT PPPPPPPPPPPP SSSSSSSSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY

```
  SSSSSSSS    AAAAAA    TTTTTTTTTT   SSSSSSSS   SSSSSSSS   SSSSSSSS  44    44    77777777
  SSSSSSSS    AAAAAA    TTTTTTTTTT   SSSSSSSS   SSSSSSSS   SSSSSSSS  44    44    77777777
SS           AA      AA     TT     SS         SS         SS           44    44          77
SS           AA      AA     TT     SS         SS         SS           44    44          77
SS           AA      AA     TT     SS         SS         SS           44    44          77
SS           AA      AA     TT     SS         SS         SS           44    44          77
  SSSSSS     AA      AA     TT       SSSSSS     SSSSSS     SSSSSS     4444444444      77
  SSSSSS     AA      AA     TT       SSSSSS     SSSSSS     SSSSSS     4444444444      77
        SS   AAAAAAAAAA     TT             SS         SS         SS         44      77
        SS   AAAAAAAAAA     TT             SS         SS         SS         44      77
        SS   AA      AA     TT             SS         SS         SS         44    77
        SS   AA      AA     TT             SS         SS         SS         44    77          ....
SSSSSSSS     AA      AA     TT     SSSSSSSS   SSSSSSSS   SSSSSSSS         44    77          ....
SSSSSSSS     AA      AA     TT     SSSSSSSS   SSSSSSSS   SSSSSSSS         44    77          ....
```

```
LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  SATSSS47 - SATS SYSTEM SERVICE TESTS  (SUCC S.C.)
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PRCVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28
0000    29  ;++
0000    30  ; FACILITY:     SATS SYSTEM SERVICE TESTS
0000    31  ;
0000    32  ; ABSTRACT:     The SATSSS47 module tests the execution of the following
0000    33  ;               VMS system services:
0000    34  ;
0000    35  ;               $SETPRV
0000    36  ;
0000    37  ; ENVIRONMENT:  User mode image,
0000    38  ;               Needs CMKRNL privilege and dynamically acquires other
0000    39  ;               privileges, as needed.
0000    40  ;
0000    41  ; AUTHOR: Larry D. Jones,                CREATION DATE: OCTOBER, 1979
0000    42  ;
0000    43  ; MODIFIED BY:
0000    44  ;
0000    45  ;       V03-001 LDJ0001         Larry D. Jones,         17-Sep-1980
0000    46  ;               Modified to conform to new build command procedures.
0000    47  ;--
0000    48  ;--
```

```
                        0000     50          .SBTTL  DECLARATIONS
                        0000     51 ;
                        0000     52 ; MACRO LIBRARY CALLS
                        0000     53 ;
                        0000     54          .LIBRARY /SYS$LIBRARY:STARLET.MLB/
                        0000     55          $JPIDEF                         ; GETJPI definitions
                        0000     56          $SHR_MESSAGES UETP,116,<<TEXT,INFO>> ; UETP$_TEXT definition
                        0000     57          $SFDEF                          ; stack frame definitions
                        0000     58          $STSDEF                         ; STS definitions
                        0000     59          $UETPDEF                        ; UETP message definitions
                        0000     60 ;
                        0000     61 ; Equated symbols
                        0000     62 ;
00000000                0000     63 WARNING          = 0                     ; warning severity value for msgs
00000001                0000     64 SUCCESS          = 1                     ; success      "      "   "   "
00000002                0000     65 ERROR            = 2                     ; error        "      "   "   "
00000003                0000     66 INFO             = 3                     ; information  "      "   "   "
00000004                0000     67 SEVERE           = 4                     ; fatal        "      "   "   "
                        0000     68 ;
                        0000     69 ;
                        0000     70 ; MACROS
                        0000     71 ;
```

```
                                                C 13
                                00000000     73          .PSECT  RODATA,RD,NOWRT,NOEXE,PAGE
                                      0000     74 ;
                                      0000     75 TEST_MOD_NAME:
        37 34 53 53 53 54 41 53 00'   0000     76          .ASCIC  /SATSSS47/                  ; needed for SATSMS message
                                   08 0000
                                      0009     77 TEST_MOD_NAME_D:
53 53 53 54 41 53 00000011'010E0000'  0009     78          .ASCID /SATSSS47/                   ; module name
                              37 34   0017
                                      0019     79 TEST_MOD_BEGIN:                             ; start end and fail messages
                 6E 69 67 65 62 00'   0019     80          .ASCIC  /begin/
                                   05 0019
                                      001F     81 TEST_MOD_SUCC:
   6C 75 66 73 73 65 63 63 75 73 00'  001F     82          .ASCIC  /successful/
                                   0A 001F
                                      002A     83 TEST_MOD_FAIL:
               64 65 6C 69 61 66 00'  002A     84          .ASCIC  /failed/
                                   06 002A
                                      0031     85 CS1:                                        ; failure messages
21 20 74 73 65 54 00000039'010E0000'  0031     86          .ASCID  \Test !AC service name !AC step !UL failed.\
6E 20 65 63 69 76 72 65 73 20 43 41   003F
70 65 74 73 20 43 41 21 20 65 6D 61   004B
2E 64 65 6C 69 61 66 20 4C 55 21 20   0057
                                      0063     87 CS2:
74 63 65 70 78 45 0000006B'010E0000'  0063     88          .ASCID  \Expected !AS = !XL received !AS = !XL\
4C 58 21 20 3D 20 53 41 21 20 64 65   0071
41 21 20 64 65 76 69 65 63 65 72 20   007D
                 4C 58 21 20 3D 20 53 0089
                                      0090     89 CS3:
74 63 65 70 78 45 00000098'010E0000'  0090     90          .ASCID  \Expected !AS!UB = !XL received !AS!UB = !XL\
20 3D 20 42 55 21 53 41 21 20 64 65   009E
64 65 76 69 65 63 65 72 20 4C 58 21   00AA
58 21 20 3D 20 42 55 21 53 41 21 20   00B6
                                   4C 00C2
                                      00C3     91 CS5:
69 20 65 64 6F 4D 000000CB'010E0000'  00C3     92          .ASCID  \Mode is !AS.\
                    2E 53 41 21 20 73 00D1
                                      00D7     93 EXP:
73 75 74 61 74 73 000000DF'010E0000'  00D7     94          .ASCID  \status\
                                      00E5     95 UM:                                         ; mode messages
       72 65 73 75 000000ED'010E0000'  00E5     96          .ASCID  \user\
                                      00F1     97 UNEXPRVCHNG:
65 70 78 65 6E 55 000000F9'010E0000'  00F1     98          .ASCID  \Unexpected privilege change.\
65 6C 69 76 69 72 70 20 64 65 74 63   00FF
      2E 65 67 6E 61 68 63 20 65 67   010B
                                      0115     99 MSGVEC:
                           00000003   0115    100          .LONG   3                          ; PUTMSG message vector
                           00741133   0119    101          .LONG   UETP$_TEXT
                           00000001   011D    102          .LONG   1
                           00000173'  0121    103          .ADDRESS MESSAGEL
                                      0125    104 SETPRV:
               56 52 50 54 45 53 00'  0125    105          .ASCIC  \SETPRV\                   ; SETPRV service name
                                   06 0125
```

```
                        012C   107 ;
                        012C   108         .SBTTL  R/W PSECT
                    00000000   109         .PSECT  RWDATA,RD,WRT,NOEXE,PAGE
                        0000   110 ;
                        0000   111 PID:
                    00000000   0000 112         .LONG   0                       ; PID for this process
                        0004   113 CURRENT_TC:
                    00000000   0004 114         .LONG   0                       ; ptr to current test case
                        0008   115         .ALIGN LONG                         ; put it on a long word boundry
                        0008   116 REG_SAVE_AREA:
                    00000044   0008 117         .BLKL   15                      ; register save area
                        0044   118 R0_SAVE:
                    00000000   0044 119         .LONG   0                       ; special case save of R0
                        0048   120 MOD_MSG_CODE:
                    007480D9   0048 121         .LONG   UETP$_SATSMS            ; test module message code for putmsg
                        004C   122 TMN_ADDR:
                    00000000'  004C 123         .ADDRESS TEST_MOD_NAME
                        0050   124 TMD_ADDR:
                    00000019'  0050 125         .ADDRESS TEST_MOD_BEGIN
                        0054   126 PRVPRT:
                          00   0054 127         .BYTE   0                       ; protection return byte for SETPRT
                        0055   128 PRIVMASK:
                00000000 00000000  0055 129         .QUAD   0                   ; priv. mask
                        005D   130 CHM_CONT:
                    00000000   005D 131         .LONG   0                       ; change mode continue address
                        0061   132 RETADR:
                    00000069   0061 133         .BLKL   2                       ; returned address's from SETPRT
                        0069   134 STATUS:
                    00000000   0069 135         .LONG   0
                        006D   136 MODE:
                    00000000   006D 137         .LONG   0                       ; current mode string pointer
                        0071   138 REG:
74 73 69 67 65 72 00000079'010E0000'  0071 139         .ASCID  \register R\
                       52 20 72 65  007F
                        0083   140 REGNUM:
                    00000000   0083 141         .LONG   0                       ; register number
                        0087   142 MSGL:
                    00000050   0087 143         .LONG   80                      ; buffer desc.
                    0000008F'  008B 144         .ADDRESS BUF
                        008F   145 BUF:
                    000000DF   008F 146         .BLKB   80
                        00DF   147 ML:
                    00000000   00DF 148         .LONG   0                       ; desc. for BUF_CHECK routine
                    000000EF'  00E3 149         .ADDRESS GETBUF+8
                        00E7   150 GETBUF:
                    00000084   00E7 151         .LONG   132
                    000000EF'  00EB 152         .ADDRESS .+4
                    00000173   00EF 153         .BLKB   132
                        0173   154 MESSAGEL:
                    00000000   0173 155         .LONG   0                       ; message desc.
                    0000008F'  0177 156         .ADDRESS BUF
                        017B   157 SERV_NAME:
                    00000000   017B 158         .LONG   0                       ; service name pointer
                        017F   159 MSGVEC1:                                     ; PUTMSG message vector
                    00000003   017F 160         .LONG   3
                    00741133   0183 161         .LONG   UETP$_TEXT
                    00000001   0187 162         .LONG   1
```

```
                    00000000  018B   163                .LONG   0
                              018F   164 GET_LIST:
                        0008  018F   165                .WORD   8
                        0400  0191   166                .WORD   JPI$_CURPRIV            ; GETJPI item list
                000001AB'     0193   167                .LONG   PRIV_LIST
                    00000000  0197   168                .LONG   0
                        0008  019B   169                .WORD   8
                        0204  019D   170                .WORD   JPI$_PROCPRIV
                000001B3'     019F   171                .LONG   PRIV_LIST+8
                    00000000  01A3   172                .LONG   0
                    00000000  01A7   173                .LONG   0
                              01AB   174 PRIV_LIST:
    00000000 00000000         01AB   175                .QUAD   0                       ; resultant CURPRIV
    00000000 00000000         01B3   176                .QUAD   0                       ; resultant PROCPRIV
                              01BB   177 PRIV_TEST:
    00000000 00000000         01BB   178                .QUAD   0                       ; privileges for SETPRV to set
                              01C3   179 PRIV_SAVE:
    00000000 00000000         01C3   180                .QUAD   0                       ; saved initial image privileges
    00000000 00000000         01CB   181                .QUAD   0                       ; saved initial process privileges
                              01D3   182 PRIV_MOD:
    00000000 00000000         01D3   183                .QUAD   0                       ; expected current image privileges
    00000000 00000000         01DB   184                .QUAD   0                       ; expected current process privileges
                              01E3   185 SET:
                              01E3   186                $SETPRV 0,0,0,PRIV_TEST         ; SETPRV parameter list
                              01F7   187
```

```
                                F 13
00000000   189          .PSECT  SATSSS47,RD,WRT,EXE,PAGE
     0000   190          .SBTTL  SATSSS47
     0000   191 ;++
     0000   192 ; FUNCTIONAL DESCRIPTION:
     0000   193 ;
     0000   194 ;       After performing some initial housekeeping, such as
     0000   195 ; printing the module begin message and acquiring needed privileges,
     0000   196 ; the system services are tested in each of their normal conditions.
     0000   197 ; Detected failures are identified and  an error message is printed
     0000   198 ; on the terminal.  Upon completion of the test a success or fail
     0000   199 ; message is printed on the terminal.
     0000   200 ;
     0000   201 ; CALLING SEQUENCE:
     0000   202 ;
     0000   203 ;       $ RUN SATSSS47  ...  (DCL COMMAND)
     0000   204 ;
     0000   205 ; INPUT PARAMETERS:
     0000   206 ;
     0000   207 ;       none
     0000   208 ;
     0000   209 ; IMPLICIT INPUTS:
     0000   210 ;
     0000   211 ;       none
     0000   212 ;
     0000   213 ; OUTPUT PARAMETERS:
     0000   214 ;
     0000   215 ;       none
     0000   216 ;
     0000   217 ; IMPLICIT OUTPUTS:
     0000   218 ;
     0000   219 ;       Messages to SYS$OUTPUT are the only output from SATSSS47.
     0000   220 ;       They are of the form:
     0000   221 ;
     0000   222 ;               %UETP-S-SATSMS, TEST MODULE SATSSS47 BEGUN ... (BEGIN MSG)
     0000   223 ;               %UETP-S-SATSMS, TEST MODULE SATSSS47 SUCCESSFUL ... (END MSG)
     0000   224 ;               %UETP-E-SATSMS, TEST MODULE SATSSS47 FAILED ... (END MSG)
     0000   225 ;               %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
     0000   226 ;
     0000   227 ; COMPLETION CODES:
     0000   228 ;
     0000   229 ;       The SATSSS47 routine terminates with a $EXIT to the
     0000   230 ;       operating system with a status code defined by UETP$_SATSMS.
     0000   231 ;
     0000   232 ; SIDE EFFECTS:
     0000   233 ;
     0000   234 ;       none
     0000   235 ;
     0000   236 ;--
     0000   237
     0000   238          TEST_START SATSSS47                     ; let the test begin
```

```
                              0000
                      0000    0000              .ENTRY SATSSS47,0
              0004'CF    D4   0002              CLRL    W^CURRENT_TC
                   00    DD   0006              PUSHL   #0
              0000'CF    DF   0008              PUSHAL  W^TPID
          00000000'GF 02 FB   000C              CALLS   #2,G^SYS$WAKE
          00000000'GF 00 FB   0013              CALLS   #0,G^SYS$HIBER
              0009'CF    7F   001A              PUSHAQ  W^TEST_MOD_NAME_D
          00000000'GF 01 FB   001E              CALLS   #1,G^SYS$SETPRN
                 0445    30   0025              BSBW    W^MOD_MSG_PRINT
        0050'CF 001F'CF  DE   0028              MOVAL   W^TEST_MOD_SUCC,W^TMD_ADDR
0048'CF    03    00  01  F0   002F              INSV    #SUCCESS,#0,#3,W^MOD_MSG_CODE
                     00  DD   0036              PUSHL   #0
            0330'CF  01  FB   0038              CALLS #1,W^REG_SAVE
                              003D      STP0:
                              003D          .SBTTL SETPRV TESTS
                              003D      ;+
                              003D      ;
                              003D      ; $SETPRV tests
                              003D      ;
                              003D      ; test _S form with a complete default parameter list
                              003D      ;
                              003D      ;-
      017B'CF   0125'CF  DE   003D          MOVAL   W^SETPRV,W^SERV_NAME          ; set service name
      006D'CF   00E5'CF  DE   0044          MOVAL   W^UM,W^MODE                   ; set the mode
                              004B          $GETJPI_S ITMLST=W^GET_LIST           ; get fresh copy of privileges
      01C3'CF   01AB'CF  7D   0060          MOVQ    W^PRIV_LIST,W^PRIV_SAVE       ; save current privileges
      01CB'CF   01B3'CF  7D   0067          MOVQ    W^PRIV_LIST+8,W^PRIV_SAVE+8   ; save process privileges
                     00  DD   006E          PUSHL   #0                            ; push a dummy parameter
            0330'CF  01  FB   0070          CALLS   #1,W^REG_SAVE                 ; save a reg snapshot
                              0075          $SETPRV_S                             ; try total default
                              0084          FAIL_CHECK SS$_NORMAL                 ; check success
          00000000'8F    DD   0084                  PUSHL   #SS$_NORMAL
            033A'CF  01  FB   008A                  CALLS   #1,W^REG_CHECK
                              008F          $GETJPI_S ITMLST=W^GET_LIST           ; get the current priv.
01AB'CF   01C3'CF    10  29   00A4          CMPC3   #16,W^PRIV_SAVE,W^PRIV_LIST   ; check for changes
                     09  13   00AC          BEQL    10$                           ; br if OK
              00F1'CF    DF   00AE          PUSHAL  W^UNEXPRVCHNG                 ; push string variable
            037C'CF  01  FB   00B2          CALLS   #1,W^PRINT_FAIL               ; print the failure
                              00B7      10$:
                              00B7      ;+
                              00B7      ;
                              00B7      ; test the PRVPRV parameter _G
                              00B7      ;
                              00B7      ;-
                              00B7          NEXT_TEST
                              00B7
                              00B7      STP1:
            0004'CF  01  D0   00B7                  MOVL    #1,W^CURRENT_TC
                     00  DD   00BC                  PUSHL   #0
            0330'CF  01  FB   00BE                  CALLS   #1,W^REG_SAVE
                              00C3          $SETPRV_G W^SET                       ; try _G with PRVPRV
                              00CC          FAIL_CHECK SS$_NORMAL                 ; check for success
          00000000'8F    DD   00CC                  PUSHL   #SS$_NORMAL
            033A'CF  01  FB   00D2                  CALLS   #1,W^REG_CHECK
01BB'CF   01C3'CF    08  29   00D7          CMPC3   #8,W^PRIV_SAVE,W^PRIV_TEST    ; check for changes
                     09  13   00DF          BEQL    20$                           ; br if OK
```

```
                                                H 13
                00F1'CF    DF  00E1   272          PUSHAL  W^UNEXPRVCHNG                 ; push string variable
            037C'CF    01   FB  00E5   273          CALLS   #1,W^PRINT_FAIL              ; print the failure
                                00EA   274 20$:
                                00EA   275 ;+
                                00EA   276 ;
                                00EA   277 ; test temp clr of one priv _G
                                00EA   278 ;
                                00EA   279 ;-
                                00EA   280          NEXT_TEST
                                00EA
                                00EA       STP2:
            0004'CF    02   D0  00EA                   MOVL    #2,W^CURRENT_TC
                       00   DD  00EF                   PUSHL   #0
            0330'CF    01   FB  00F1                   CALLS   #1,W^REG_SAVE
      01BB'CF   01C3'CF   7D  00F6   281          MOVQ    W^PRIV_SAVE,W^PRIV_TEST      ; get current image priv.
      01D3'CF   01C3'CF   7D  00FD   282          MOVQ    W^PRIV_SAVE,W^PRIV_MOD       ; make a copy of the priv.
                01E7'CF   D4  0104   283          CLRL    W^SET+SETPRV$_ENBFLG         ; set for disable
                01F3'CF   D4  0108   284          CLRL    W^SET+SETPRV$_PRVPRV         ; disable previous priv
      01EB'CF   01BB'CF   DE  010C   285          MOVAL   W^PRIV_TEST,W^SET+SETPRV$_PRVADR ; set priv. address
52    01BB'CF   1F   00   EA  0113   286          FFS     #0,#31,W^PRIV_TEST,R2        ; find a priv
                01BB'CF   D4  011A   287          CLRL    W^PRIV_TEST                  ; clear off a space to work
01BB'CF   01   52   01   F0  011E   288          INSV    #1,R2,#1,W^PRIV_TEST         ; set a bit for the priv to remove
01D3'CF   01   52   00   F0  0125   289          INSV    #0,R2,#1,W^PRIV_MOD          ; set expected results
                       00   DD  012C   290          PUSHL   #0                           ; push a dummy parameter
            0330'CF    01   FB  012E   291          CALLS   #1,W^REG_SAVE                ; save a register snapshot
                                0133   292          $SETPRV_G W^SET                      ; try _G
                                013C   293          FAIL_CHECK SS$_NORMAL                ; check results
             00000000'8F   DD  013C                   PUSHL   #SS$_NORMAL
            033A'CF    01   FB  0142                   CALLS   #1,W^REG_CHECK
                                0147   294          $GETJPI_S ITMLST=W^GET_LIST          ; get new priv.
01AB'CF   01D3'CF   08   29  015C   295          CMPC3   #8,W^PRIV_MOD,W^PRIV_LIST    ; check the results
                       09   13  0164   296          BEQL    30$                          ; br if OK
                00F1'CF    DF  0166   297          PUSHAL  W^UNEXPRVCHNG                ; push str var
            037C'CF    01   FB  016A   298          CALLS   #1,W^PRINT_FAIL              ; print the failure
                                016F   299 30$:
                                016F   300 ;+
                                016F   301 ;
                                016F   302 ; test temp adding of one priv _S
                                016F   303 ;
                                016F   304 ;-
                                016F   305          NEXT_TEST
                                016F
                                016F       STP3:
            0004'CF    03   D0  016F                   MOVL    #3,W^CURRENT_TC
                       00   DD  0174                   PUSHL   #0
            0330'CF    01   FB  0176                   CALLS   #1,W^REG_SAVE
      01D3'CF   01AB'CF   7D  017B   306          MOVQ    W^PRIV_LIST,W^PRIV_MOD       ; save a copy of the privs
      01BB'CF   01AB'CF   7D  0182   307          MOVQ    W^PRIV_LIST,W^PRIV_TEST      ;
52    01BB'CF   1F   00   EB  0189   308          FFC     #0,#31,W^PRIV_TEST,R2        ; find a missing priv
                01BB'CF   7C  0190   309          CLRQ    W^PRIV_TEST                  ; clean out the bits
01BB'CF   01   52   01   F0  0194   310          INSV    #1,R2,#1,W^PRIV_TEST         ; enable that priv.
01D3'CF   01   52   01   F0  019B   311          INSV    #1,R2,#1,W^PRIV_MOD          ; make expected results
                       00   DD  01A2   312          PUSHL   #0                           ; push a dummy parameter
            0330'CF    01   FB  01A4   313          CALLS   #1,W^REG_SAVE                ; save a reg snapshot
                                01A9   314          $SETPRV_S ENBFLG=#1,-
                                01A9   315                    PRVADR=W^PRIV_TEST     ; try _S
                                01BA   316          FAIL_CHECK SS$_NORMAL                ; check success
```

```
              00000000'8F    DD  01BA                      PUSHL   #SS$_NORMAL
              033A'CF   01   FB  01C0                      CALLS   #1,W^REG_CHECK
                             01C5          $GETJPI_S ITMLST=W^GET_LIST             ; get current priv
   01AB'CF    01D3'CF   08   29  01DA          CMPC3   #8,W^PRIV_MOD,@^PRIV_LIST   ; check for the change
                        09   13  01E2          BEQL    40$                         ; br if OK
                 00F1'CF     DF  01E4          PUSHAL  W^UNEXPRVCHNG               ; push str var
              037C'CF   01   FB  01E8          CALLS   #1,W^PRINT_FAIL             ; print the failure
                             01ED  40$:
                             01ED  ;+
                             01ED  ;
                             01ED  ; test the perm clearing of one privilege _G
                             01ED  ;
                             01ED  ;-
                             01ED          NEXT_TEST
                             01ED
                             01ED      STP4:
              0004'CF   04   D0  01ED                      MOVL    #4,W^CURRENT_TC
                        00   DD  01F2                      PUSHL   #0
              0330'CF   01   FB  01F4                      CALLS   #1,W^REG_SAVE
       01BB'CF    01CB'CF    7D  01F9          MOVQ    W^PRIV_SAVE+8,W^PRIV_TEST   ; get process priv.
       01D3'CF    01CB'CF    7D  0200          MOVQ    W^PRIV_SAVE+8,W^PRIV_MOD
   52  01BB'CF   1F     00   EA  0207          FFS     #0,#31,W^PRIV_TEST,R2       ; find a priv
                 01BB'CF     D4  020E          CLRL    W^PRIV_TEST                 ; clear off a space to work
   01BB'CF    01   52   01   F0  0212          INSV    #1,R2,#1,W^PRIV_TEST        ; set a bit for the priv to remove
   01D3'CF    01   52   00   F0  0219          INSV    #0,R2,#1,W^PRIV_MOD         ; set expected results
              01EF'CF   01   D0  0220          MOVL    #1,W^SET+SETPRV$_PRMFLG     ; set the perm flag
                        00   DD  0225          PUSHL   #0                          ; push a dummy parameter
              0330'CF   01   FB  0227          CALLS   #1,W^REG_SAVE               ; save a reg snapshot
                             022C          $SETPRV_G W^SET                         ; try _G
                             0235          FAIL_CHECK SS$_NORMAL                   ; check for success
              00000000'8F    DD  0235                      PUSHL   #SS$_NORMAL
              033A'CF   01   FB  023B                      CALLS   #1,W^REG_CHECK
                             0240          $GETJPI_S ITMLST=W^GET_LIST             ; get current priv.
   01B3'CF    01D3'CF   08   29  0255          CMPC3   #8,W^PRIV_MOD,@^PRIV_LIST+8 ; check the priv.'s
                        09   13  025D          BEQL    60$                         ; br if OK
                 00F1'CF     DF  025F          PUSHAL  W^UNEXPRVCHNG               ; push string variable
              037C'CF   01   FB  0263          CALLS   #1,W^PRINT_FAIL             ; print the failure
                             0268  60$:
                        00   DD  0268          PUSHL   #0                          ; push a dummy parameter
              0330'CF   01   FB  026A          CALLS   #1,W^REG_SAVE               ; save a reg snapshot
                             026F          $SETPRV_S ENBFLG=#1,-
                             026F                  PRVADR=W^PRIV_SAVE+8,-
                             026F                  PRMFLG=#1                       ; reset perm priv to original
                             0280          FAIL_CHECK SS$_NORMAL                   ; check for failure
              00000000'8F    DD  0280                      PUSHL   #SS$_NORMAL
              033A'CF   01   FB  0286                      CALLS   #1,W^REG_CHECK
                             028B  ;+
                             028B  ;
                             028B  ; test perm add one priv _S
                             028B  ;
                             028B  ;-
                             028B          NEXT_TEST
                             028B
                             028B      STP5:
              0004'CF   05   D0  028B                      MOVL    #5,W^CURRENT_TC
                        00   DD  0290                      PUSHL   #0
              0330'CF   01   FB  0292                      CALLS   #1,W^REG_SAVE
```

```
       01D3'CF   01B3'CF    7D  0297       MOVQ    W^PRIV_LIST+8,W^PRIV_MOD        ; save a copy of the privs
       01BB'CF   01B3'CF    7D  029E       MOVQ    W^PRIV_LIST+8,W^PRIV_TEST
  52   01BB'CF    1F    00  EB  02A5       FFC     #0,#31,W^PRIV_TEST,R2           ; find a missing priv
            01BB'CF         D4  02AC       CLRL    W^PRIV_TEST                     ; clean up the bits
 01BB'CF   01    52    01   F0  02B0       INSV    #1,R2,#1,W^PRIV_TEST            ; add the missing priv
 01D3'CF   01    52    01   F0  02B7       INSV    #1,R2,#1,W^PRIV_MOD             ; make expected results
                       00   DD  02BE       PUSHL   #0                              ; push a dummy parameter
          0330'CF      01   FB  02C0       CALLS   #1,W^REG_SAVE                   ; save a reg snapshot
                                02C5       $SETPRV_S ENBFLG=#1,-
                                02C5               PRVADR=W^PRIV_TEST,-
                                02C5               PRMFLG=#1                       ; try _S
                                02D6       FAIL_CHECK SS$_NORMAL                   ; check for success
          00000000'8F       DD  02D6               PUSHL   #SS$_NORMAL
          033A'CF      01   FB  02DC               CALLS   #1,W^REG_CHECK
                                02E1       $GETJPI_S ITMLST=W^GET_LIST             ; get the current priv.
 01D3'CF   01B3'CF     08   29  02F6       CMPC3   #8,W^PRIV_LIST+8,W^PRIV_MOD     ; check for change
                       09   13  02FE       BEQL    50$                             ; br if OK
             00F1'CF        DF  0300       PUSHAL  W^UNEXPRVCHNG                   ; push the str variable
          037C'CF      01   FB  0304       CALLS   #1,W^PRINT_FAIL                 ; print the failure
                                0309  50$:
                                0309       TEST_END
             0050'CF        DD  0309               PUSHL   W^TMD_ADDR
             004C'CF        DD  030D               PUSHL   W^TMN_ADDR
                  02        DD  0311               PUSHL   #2
             0048'CF        DD  0313               PUSHL   W^MOD_MSG_CODE
       00000000'GF     04   FB  0317               CALLS   #$$T1,G^LIB$SIGNAL
 0048'CF   01    1C    01   F0  031E               INSV    #1,#STS$V_INHIB_MSG,#1,W^MOD_MSG_CODE
             0048'CF        DD  0325               PUSHL   W^MOD_MSG_CODE
       00000000'GF     01   FB  0329               CALLS   #1,G^SYS$EXIT
```

```
                                0330   379          .SBTTL REG_SAVE
                                0330   380 ;++
                                0330   381 ; FUNCTIONAL DESCRIPTION:
                                0330   382 ;       Subroutine to save R2-R11 in the register save location.
                                0330   383 ;
                                0330   384 ; CALLING SEQUENCE:
                                0330   385 ;       PUSHL   #0                      ; save a dummy parameter
                                0330   386 ;       CALLS   #1,W^REG_SAVE   ; save R2-R11
                                0330   387 ;
                                0330   388 ; INPUT PARAMETERS:
                                0330   389 ;       NONE
                                0330   390 ;
                                0330   391 ; OUTPUT PARAMETERS:
                                0330   392 ;       NONE
                                0330   393 ;
                                0330   394 ;--
                                0330   395
                                0330   396 REG_SAVE:
                           0FFC 0330   397          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
     0008'CF   14 AD   28    28 0332   398          MOVC3   #4*10,^X14(FP),W^REG_SAVE_AREA  ; save the registers in the program
                             04 0339   399          RET
                                033A   400          .SBTTL  REG_CHECK
                                033A   401 ;++
                                033A   402 ; FUNCTIONAL DESCRIPTION:
                                033A   403 ;       Subroutine to test R0 & R2-R11 for proper content after a service
                                033A   404 ;       execution. A snapshot is taken by the REG_SAVE routine at the
                                033A   405 ;       beginning of each step and this routine is executed after the
                                033A   406 ;       services have been executed.
                                033A   407 ;
                                033A   408 ; CALLING SEQUENCE:
                                033A   409 ;       PUSHL   #SS$_XXXXXX     ; push expected R0 contents
                                033A   410 ;       CALLS   #1,W^REG_CHECK  ; execute this routine
                                033A   411 ;
                                033A   412 ; INPUT PARAMETERS:
                                033A   413 ;       expected R0 contents on the stack
                                033A   414 ;
                                033A   415 ; OUTPUT PARAMETERS:
                                033A   416 ;       possible error messages printed using $PUTMSG
                                033A   417 ;
                                033A   418 ;--
                                033A   419
                                033A   420 REG_CHECK:
                           0FFC 033A   421          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          50   04 AC     D1    033C   422          CMPL    4(AP),R0                        ; is this the right fail code?
                       0E   13 0340   423          BEQL    10$                             ; br if yes
                       50   DD 0342   424          PUSHL   R0                              ; push received data
                    04 AC   DD 0344   425          PUSHL   4(AP)                           ; push expected data
                 00D7'CF    DF 0347   426          PUSHAL  W^EXP                           ; push the string variable
               037C'CF      03 FB 034B   427          CALLS   #3,W^PRINT_FAIL                 ; print the error message
                                0350   428 10$:
     0008'CF   14 AD   28    29 0350   429          CMPC3   #4*10,^X14(FP),W^REG_SAVE_AREA  ; check all but R0
                       22   13 0357   430          BEQL    20$                             ; br if O.K.
56   53   00000008'8F      C3 0359   431          SUBL3   #REG_SAVE_AREA,R3,R6            ; calculate the register number
                    56   04 C6 0361   432          DIVL2   #4,R6
                 7E 56   02 81 0364   433          ADDB3   #^X2,R6,-(SP)                   ; set number past R0-R1 and save
                    51   03 CA 0368   434          BICL2   #3,R1                           ; backup to register boundrys
                    53   03 CA 036B   435          BICL2   #3,R3
```

```
                                                                L 13
                61   DD  036E   436          PUSHL   (R1)                          ; push received data
                63   DD  0370   437          PUSHL   (R3)                          ; push expected data
          0071'CF    DF  0372   438          PUSHAL  W^REG                         ; set string pntr param.
037C'CF    04        FB  0376   439          CALLS   #4,W^PRINT_FAIL               ; print the error message
                     0378   440 20$:
                04   0378   441          RET
                     037C   442          .SBTTL  PPINT_FAIL
                     037C   443 ;++
                     037C   444 ; FUNCTIONAL DESCRIPTION:
                     037C   445 ;       Subroutine to report failures using $PUTMSG
                     037C   446 ;
                     037C   447 ; CALLING SEQUENCE:
                     037C   448 ; Mode  #1       PUSHL EXPECTED  Mode    #2      PUSHL REG_NUMBER
                     037C   449 ;                PUSHL RECEIVED                  PUSHL EXPECTED
                     037C   450 ;                PUSHAL STRING_VAR               PUSHL RECEIVED
                     037C   451 ;                CALLS #3,W^PRINT_FAIL           PUSHAL STRING_VAR
                     037C   452 ;                                                CALLS #4,W^PRINT_FAIL
                     037C   453 ; Mode  #3       PUSHAL STRING_VAR
                     037C   454 ;                CALLS #1,W^PRINT_FAIL
                     037C   455 ;
                     037C   456 ; INPUT PARAMETERS:
                     037C   457 ;       listed above
                     037C   458 ;
                     037C   459 ; OUTPUT PARAMETERS:
                     037C   460 ;       an error message is printed using $PUTMSG
                     037C   461 ;
                     037C   462 ;--
                     037C   463
                     037C   464 PRINT_FAIL:
                003C 037C   465          .WORD   ^M<R2,R3,R4,R5>
                     037E   466          $FAO_S  W^CS1,W^MESSAGEL,W^MSGL,#TEST_MOD_NAME,W^SERV_NAME,W^CURRENT_TC
                     039F   467          $PUTMSG_S W^MSGVEC                    ; print the message
             04  6C  91   03B0   468          CMPB    (AP),#4                       ; is this a register message?
                 26  13   03B3   469          BEQL    10$                           ; br if yes
             01  6C  91   03B5   470          CMPB    (AP),#1                       ; is this just a message?
                 48  13   03B8   471          BEQL    20$                           ; br if yes
                     03BA   472          $FAO_S  W^CS2,W^MESSAGEL,W^MSGL,4(AP),8(AP),4(AP),12(AP)
                 40  11   03D9   473          BRB     30$                           ; goto output message
                     03DB   474 10$:
                     03DB   475          $FAO_S  W^CS3,W^MESSAGEL,W^MSGL,4(AP),16(AP),8(AP),4(AP),16(AP),12(AP)
                 19  11   0400   476          BRB     30$                           ; goto output message
                     0402   477 20$:
      018B'CF    04 AC  D0   0402   478          MOVL    4(AP),W^MSGVEC1+12            ; save string address
                     0408   479          $PUTMSG_S W^MSGVEC1                   ; print the message
                 11  11   0419   480          BRB     40$                           ; skip the other message
                     041B   481 30$:
                     041B   482          $PUTMSG_S W^MSGVEC                    ; print the message
                     042C   483 40$:
        0440'CF  00  FB   042C   484          CALLS   #0,W^MODE_ID                  ; identify the mode
    0050'CF  002A'CF  DE   0431   485          MOVAL   W^TEST_MOD_FAIL,W^TMD_ADDR    ; set failure message address
0048'CF  03  00   02  F0   0438   486          INSV    #ERROR,#0,#3,W^MOD_MSG_CODE   ; set severity code
                 04  043F   487          RET
```

```
                     0440   489          .SBTTL  MODE_ID
                     0440   490 ;++
                     0440   491 ; FUNCTIONAL DESCRIPTION:
                     0440   492 ;       Subroutine to identify the mode that an exit handler is in.
                     0440   493 ;
                     0440   494 ; CALLING SEQUENCE:
                     0440   495 ;       CALLS   #0,W^MODE_ID
                     0440   496 ;
                     0440   497 ; INPUT PARAMETERS:
                     0440   498 ;       MODE contains an address pointing to an ascii string desc.
                     0440   499 ;       of the current CPU mode.
                     0440   500 ;
                     0440   501 ; OUTPUT PARAMETERS:
                     0440   502 ;       NONE
                     0440   503 ;
                     0440   504 ;--
                     0440   505
                     0440   506 MODE_ID:
                003C 0440   507          .WORD   ^M<R2,R3,R4,R5>
                     0442   508          $FAO_S  W^CS5,W^MESSAGEL,W^MSGL,MODE ; format the error message
                     045B   509          $PUTMSG_S W^MSGVEC                   ; print the mode message
                  04 046C   510          RET
```

```
                     046D   513 MOD_MSG_PRINT:
                     046D   514 ;
                     046D   515 ;  ****************************************************************************
                     046D   516 ;  *                                                                          *
                     046D   517 ;  *  PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES                  *
                     046D   518 ;  *       (USING THE PUTMSG MACRO).                                          *
                     046D   519 ;  *                                                                          *
                     046D   520 ;  ****************************************************************************
                     046D   521 ;
                     046D   522          PUTMSG  <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> ; PRINT MSG
                  05 0488   523          RSB                                         ; ... AND RETURN TO CALLER
                     0489   524 ;
                     0489   525 CHMRTN:
                     0489   526 ; ****************************************************************************
                     0489   527 ; *                                                                          *
                     0489   528 ; *     CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER               *
                     0489   529 ; *     A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED                  *
                     0489   530 ; *     BY THE MODE MACRO ('TO' OPTION).  IT MERELY DOES                     *
                     0489   531 ; *     A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS                    *
                     0489   532 ; *     THE EFFECT OF RETURNING TO THE END OF THE MODE                       *
                     0489   533 ; *     MACRO EXPANSION.                                                     *
                     0489   534 ; *                                                                          *
                     0489   535 ; ****************************************************************************
                     0489   536 ;
                0000 0489   537          .WORD   0                                   ; ENTRY MASK
    0000005D'FF   17 048B   538          JMP     @CHM_CONT                           ; RETURN TO MODE MACRO IN NEW MODE
                     0491   539 ;
                     0491   540 ; *     RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
                     0491   541 ;
                     0491   542 TEST_END:
                     0491   543          .END    SATSSS47
```

```
$$ARGS                  =  00000004
$$T1                    =  00000004
$$T2                    =  00000004
BUF                        0000008F  R      03
CHMRTN                     00000489  R      04
CHM_CONT                   0000005D  R      03
CS1                        00000031  R      02
CS2                        00000063  R      02
CS3                        00000090  R      02
CS5                        000000C3  R      02
CURRENT_TC                 00000004  R      03
ERROR                   =  00000002
EXP                        000000D7  R      02
GETBUF                     000000E7  R      03
GET_LIST                   0000018F  R      03
INFO                    =  00000003
JPI$_CURPRIV            =  00000400
JPI$_PROCPRIV           =  00000204
LIB$SIGNAL                 ********    X    04
MESSAGEL                   00000173  R      03
ML                         000000DF  R      03
MODE                       0000006D  R      03
MODE_ID                    00000440  R      04
MOD_MSG_CODE               00000048  R      03
MOD_MSG_PRINT              0000046D  R      04
MSGL                       00000C87  R      03
MSGVEC                     00000115  R      02
MSGVEC1                    0000017F  R      03
PRINT_FAIL                 0000037C  R      04
PRIVMASK                   00000055  R      03
PRIV_LIST                  000001AB  R      03
PRIV_MOD                   000001D3  R      03
PRIV_SAVE                  000001C3  R      03
PRIV_TEST                  000001BB  R      03
PRVPRT                     00000054  R      03
RO_SAVE                    00000044  R      03
REG                        00000071  R      03
REGNUM                     00000083  R      03
REG_CHECK                  0000033A  R      04
REG_SAVE                   00000330  R      04
REG_SAVE_AREA              00000008  R      03
RETADR                     00000061  R      03
SATSSS47                   00000000  RG     04
SERV_NAME                  0000017B  R      03
SET                        000001E3  R      03
SETPRV                     00000125  R      02
SETPRV$_ENBFLG          =  00000004
SETPRV$_NARGS           =  00000004
SETPRV$_PRMFLG          =  0000000C
SETPRV$_PRVADR          =  00000008
SETPRV$_PRVPRV          =  00000010
SEVERE                  =  00000004
SHR$K_SHRDEF            =  00000001
SHR$_TEXT               =  00001130
SS$_NORMAL                 ********    X    04
STATUS                     00000069  R      03
STEP                    =  00000005
STP0                       0000003D  R      04
STP1                       000000B7  R      04
STP2                       000000EA  R      04
STP3                       0000016F  R      04
STP4                       000001ED  R      04
STP5                       0000028B  R      04
STS$V_INHIB_MSG         =  0000001C
SUCCESS                 =  00000001
SYS$EXIT                   ********   GX    04
SYS$FAO                    ********    X    04
SYS$GETJPI                 ********   GX    04
SYS$HIBER                  ********   GX    04
SYS$PUTMSG                 ********   GX    04
SYS$SETPRN                 ********   GX    04
SYS$SETPRV                 ********   GX    04
SYS$WAKE                   ********   GX    04
TEST_END                   00000491  R      04
TEST_MOD_BEGIN             00000019  R      02
TEST_MOD_FAIL              0000002A  R      02
TEST_MOD_NAME              00000000  R      02
TEST_MOD_NAME_D            00000009  R      02
TEST_MOD_SUCC              0000001F  R      02
TMD_ADDR                   00000050  R      03
TMN_ADDR                   0000004C  R      03
TPID                       00000000  R      03
UETP$_SATSMS            =  007480D9
UETP$_TEXT              =  00741133
UM                         000000E5  R      02
UNEXPRVCHNG                000000F1  R      02
WARNING                 =  00000000
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RODATA	0000012C (300.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC PAGE
RWDATA	000001F7 (503.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC PAGE
SATSSS47	00000491 (1169.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC PAGE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.08	00:00:00.37
Command processing	137	00:00:00.80	00:00:04.21
Pass 1	263	00:00:07.19	00:00:13.92
Symbol table sort	0	00:00:00.52	00:00:00.59
Pass 2	119	00:00:01.92	00:00:03.38
Symbol table output	10	00:00:00.08	00:00:00.09
Psect synopsis output	3	00:00:00.02	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	569	00:00:10.61	00:00:22.60

The working set limit was 1350 pages.
42200 bytes (83 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 378 non-local and 12 local symbols.
543 source lines were read in Pass 1, producing 25 object records in Pass 2.
38 pages of virtual memory were used to define 34 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	21
_255DUA28:[SHRLIB]UETP.MLB;1	10
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	0
TOTALS (all libraries)	31

582 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SATSSS47/OBJ=OBJ$:SATSSS47 MSRC$:SATSSS47/UPDATE=(ENH$:SATSSS47)+EXECML$/LIB+SHRLIB$:UETP/LIB
```